ФГБОУ ВПО «Воронежский государственный
технический университет»

Кафедра систем информационной безопасности

192-2015

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам № 5–8 по дисциплинам
«Защита в операционных системах»,
«Безопасность операционных систем»
для студентов специальностей
090301 «Компьютерная безопасность»,
090303 «Информационная безопасность
автоматизированных систем»
очной формы обучения

Воронеж 2015

Составители: д-р техн. наук А. Ю. Савинков, Н. А. Ленков, канд. техн. наук В. Н. Деревянко

УДК 004.056.5

Методические указания к лабораторным работам № 5–8 по дисциплинам «Защита в операционных системах», «Безопасность операционных систем» для студентов специальностей 090301 «Компьютерная безопасность», 090303 «Информационная безопасность автоматизированных систем» очной формы обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. А. Ю. Савинков, Н. А. Ленков, В. Н. Деревянко. Воронеж, 2015. 40 с.

Методические указания предназначены для изучения принципов работы операционных систем, их администрирования и обеспечения безопасности, а также для закрепления лекционного материала на практике.

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2013 и содержатся в файле Савинков_ЛР_ЗОС+БОС_5-8.pdf.

Табл. 2. Ил. 1. Библиогр.: 6 назв.

Рецензент д-р техн. наук, проф. А. Г. Остапенко

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. А. Г. Остапенко

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

Лабораторная работа № 5
Работа с виртуальными машинами

Цель работы: Изучение основных понятий о виртуальных машинах для их практического применения.

Теоретические сведения

1. Определение и понятие виртуальной машины

Чтобы построить полный взгляд на виртуальные машины, разберем для начала, а что такое виртуальная машина?

Виртуальная машина — программная или аппаратная среда, исполняющая некоторый код (например, байт-код, шитый код, p-код или машинный код реального процессора), или спецификация такой системы (например: «виртуальная машина языка программирования Си»).

Для сравнения приведем несколько других определенней, а именно: Виртуальная машина — это полностью изолированный программный контейнер, способный выполнять собственную операционную систему и приложения, как физический компьютер. Виртуальная машина работает абсолютно так же, как физический компьютер, и содержит собственные виртуальные (т.е. программные) ЦП, ОЗУ, жесткий диск и сетевую интерфейсную карту (NIC).

Проще говоря, виртуальная машина – это программа, которую вы запускаете из своей операционной системы. Программа эмулирует реальную машину. На виртуальные машины, как и на реальные, можно ставить операционные системы. У неё есть BIOS, отведенное место на вашем жестком диске, сетевые адаптеры для соединения с реальной машиной, сетевыми ресурсами или другими виртуальными машинами.

2. Преимущества и недостатки виртуальных машин

Приведем несколько преимуществ использования виртуальных машин:

1. Приведу самый просто пример. Нынче, как мы знаем, вышли новые операционные системы. Windows Vista и Windows 7. И как многие из вас убедились, некоторые приложения, в частности игры, на них не работают. Так в чём проблема? Когда можно установить виртуальную машину с, допустим, операционной системой Windows XP. И всё прекрасно будет работать.
2. Второй пункт можно отнести к злобным хакерам или просто к компьютерным хулиганам. Имеется в виду, что на виртуальной машине вы можете спокойно написать вирус или вредоносное программное обеспечение, которое сможет повредить вам лишь гостевую операционную систему виртуальной машины.
3. Третий пункт можно было отнести ко второму. А именно то, что на виртуальную машину вы можете ставить любое ПО, не опасаясь чего-либо. Вы можете экспериментировать с различными настройками и прочее.
4. Ну и одно из самых главных это то, что вы можете легко изучать новые операционные системы, не стирая свою старую.

Это конечно далеко не все преимущества виртуальных машин. Каждый пользователь может сам придумать, для чего ему нужна виртуальная машина.

Перед возможностью установки нескольких хостовых операционных систем на один компьютер с их раздельной загрузкой, виртуальные машины имеют следующие неоспоримые преимущества:

1. Возможность работать одновременно в нескольких системах, осуществлять сетевое взаимодействие между ними.
2. Возможность сделать «снимок» текущего состояния системы и содержимого дисков одним кликом

мыши, а затем в течение очень короткого промежутка времени вернуться в исходное состояние.

3. Простота создания резервной копии операционной системы (не надо создавать никаких образов диска, всего лишь требуется скопировать папку с файлами виртуальной машины).

4. Возможность иметь на одном компьютере неограниченное число виртуальных машин с совершенно разными операционными системами и их состояниями.

5. Отсутствие необходимости перезагрузки для переключения в другую операционную систему.

Тем не менее, несмотря на все преимущества, виртуальные машины также имеют и свои недостатки:

1. Потребность в наличии достаточных аппаратных ресурсов для функционирования нескольких операционных систем одновременно.

2. Операционная система работает несколько медленнее в виртуальной машине, нежели на «голом железе». Однако, в последнее время показатели производительности гостевых систем значительно приблизились к показателям физических ОС (в пределах одних и тех же ресурсов), и вскоре, за счет улучшения технологий реализации виртуальных машин, производительность гостевых систем практически будет равна реальным.

3. Существуют методы определения того, что программа запущена в виртуальной машине (в большинстве случаев, производители систем виртуализации сами предоставляют такую возможность). Вирусописатели и распространители вредоносного программного обеспечения, конечно же, в курсе этих методов и в последнее время включают в свои программы функции обнаружения факта запуска в виртуальной машине, при этом никакого ущерба вредоносное ПО гостевой системе не причиняет.

4. Различные платформы виртуализации пока не поддерживают полную виртуализацию всего аппаратного

обеспечения и интерфейсов. В последнее время количество поддерживаемого аппаратного обеспечения стремительно растет у всех производителей платформ виртуализации. Помимо основных устройств компьютера, уже поддерживаются сетевые адаптеры, аудиоконтроллеры, интерфейс USB 2.0, котроллеры портов COM и LPT и приводы CD-ROM. Но хуже всего обстоят дела с виртуализацией видеоадаптеров и поддержкой функций аппаратного ускорения трехмерной графики.

Все недостатки в принципе можно решить, да и преимущества виртуальных машин перевешивают их недостатки. Именно поэтому виртуализация сейчас продвигается семимильными шагами вперёд. А пользователи находят всё больше и больше причин их использовать.

3. Архитектура виртуальных машин

Виртуализация один из важных инструментов разработки компьютерных систем, а сами виртуальные машины используются в самых разных областях.

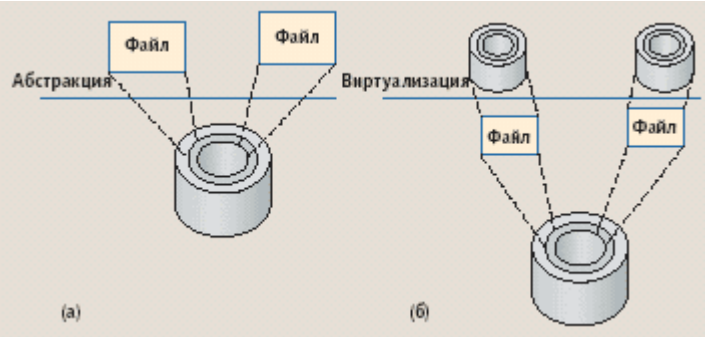
Виртуальные машины разрабатываются большим количеством специалистов, преследующих самые разные цели, и в этой области существует не так уж много общепринятых концепций. Поэтому лучше всего будет рассмотреть понятие виртуализации и всё разнообразие архитектур виртуальных машин в единой перспективе.

3.1. Абстракция и виртуализация

Компьютерные системы разрабатываются по определенной иерархии и имеют хорошо определенные интерфейсы, из-за чего они и продолжают развиваться. Использование таких интерфейсов облегчает независимую разработку аппаратных и программных подсистем силами разных групп специалистов. Абстракции скрывают детали

реализации нижнего уровня, уменьшая сложность процесса проектирования [1].

На рисунке приведен пример абстракции в применении к дисковой памяти. Операционная система абстрагируется от тонкостей адресации на жестком диске, от его секторов и дорожек, чтобы для прикладной программы диск выглядел как набор файлов переменного размера. Опираясь на эту абстракцию, «прикладные» программисты могут создавать файлы, записывать и читать данные, не зная устройства и физической организации жесткого диска [1].



Абстракция и виртуализация в применение
к дисковой памяти

Концепция архитектуры системы команд компьютера (instruction set architecture, ISA) наглядно иллюстрирует преимущества хорошо определенных интерфейсов. Они позволяют разрабатывать взаимодействующие компьютерные подсистемы не только в разных организациях, но и в разные периоды, иногда разделенные годами. Например, Intel и AMD создают микропроцессоры с системой команд IA-32 (x86), в то время как разработчики Microsoft пишут программное обеспечение, которое компилируется в эту систему команд. Поскольку обе стороны соблюдают спецификацию ISA, можно ожидать, что программное обеспечение будет правильно

выполняться любым ПК на базе микропроцессора с архитектурой IA-32.

К сожалению, хорошо определенные интерфейсы имеют и недостатки. Подсистемы и компоненты, разработанные по спецификациям разных интерфейсов, не способны взаимодействовать друг с другом. Например, приложения, распространяемые в двоичных кодах, привязаны к определенной ISA и зависят от конкретного интерфейса к операционной системе. Несовместимость интерфейсов может стать сдерживающим фактором, особенно в мире компьютерных сетей, в котором свободное перемещение программ столь же необходимо, как и перемещение данных.

Виртуализация позволяет обойти эту несовместимость. Виртуализация системы или компонента (например, процессора, памяти или устройства ввода/вывода) на конкретном уровне абстракции отображает его интерфейс и видимые ресурсы на интерфейс и ресурсы реальной системы. Следовательно, реальная система выступает в роли другой, виртуальной системы или даже нескольких виртуальных систем.

В отличие от абстракции, виртуализация не всегда нацелена на упрощение или сокрытие деталей. В примере на рисунке виртуализация позволяет преобразовать один большой диск в два меньших виртуальных диска, каждый из которых имеет собственные секторы и дорожки [1]. При отображении виртуальных дисков на реальный программные средства виртуализации используют абстракцию файла как промежуточный шаг. Операция записи на виртуальный диск преобразуется в операцию записи в файл (и, следовательно, в операцию записи на реальный диск). Отметим, что в данном случае никакого абстрагирования не происходит — уровень детализации интерфейса виртуального диска (адресация секторов и дорожек) ничем не отличается от уровня детализации реального диска.

3.2. Процессные и системные виртуальные машины

Понятия пошли от того, что система и процесс видят машину по-разному, поэтому и виртуальные машины бывают процессные и системные [1].

Процессная виртуальная машина — это виртуальная платформа для выполнения отдельного процесса. Она предназначена для поддержки процесса, создаётся при его активации и «умирает» после его окончания. Системная виртуальная машина – полнофункциональная, постоянно действующая системная среда, служащая для поддержки операционной системы вместе с большим количеством её пользовательских процессов; она обеспечивает «гостевой» операционной системе доступ к виртуальным аппаратным средствам, в том числе к процессору и памяти, устройствам ввода/вывода, а иногда — и к графическому интерфейсу.

Процесс или система, которые выполняются на виртуальной машине, называются гостем, платформа, поддерживающая виртуальную машину, — хостом. Программное обеспечение, реализующее процессную виртуальную машину, называют рабочей средой, а программное обеспечение виртуализации системной виртуальной машины – монитором виртуальной машины.

Процессные виртуальные машины создают среды ABI и API для пользовательских приложений, что позволяет в многозадачном режиме осуществлять репликацию операционной среды, эмулировать систему команд, оптимизировать код или выполнять программы на языках высокого уровня.

Системная виртуальная машина обеспечивает полнофункциональную среду, в которой могут сосуществовать операционная система и несколько процессов, относящихся к разным пользователям [1]. С помощью них одна аппаратная платформа может поддерживать несколько гостевых операционных систем одновременно.

3.3. Типы виртуализаций

Рассмотрим основные типы виртуализации различных компонент ИТ — инфраструктуры [1].

1. Виртуализация операционной системы. Является наиболее распространенной в данный момент формой виртуализации. Виртуальная операционная система (виртуальная машина) представляет собой, как правило, совмещение нескольких операционных систем, функционирующих на одной аппаратной основе. Каждая из виртуальных машин управляется отдельно при помощи VMM (Virtual Machine Manager). Лидерами в области поставок решений для виртуализации информационных систем являются Microsoft, AMD, Intel и VMware.

2. Виртуализация серверов приложений. Под данным процессом виртуализации понимают процесс интеллектуальной балансировки нагрузки. Балансировщик нагрузки управляет несколькими веб — серверами и приложениями, как единой системой, пользователь, при этом, «видит» только один сервер, который, фактически, предоставляет функционал нескольких серверов.

3. Виртуализация приложений. Под виртуализацией приложений следует понимать использование программных решений в рамках изолированной виртуальной среды (более подробно виртуализация приложений будет рассмотрена в последующих лекциях).

4. Виртуализация сети. Представляет собой объединение аппаратных и программных ресурсов в единую виртуальную сеть. Выделяют внутреннюю виртуализацию сети — создающую виртуальную сеть между виртуальными машинами одной системы, и внешнюю — объединяющую несколько сетей в одну виртуальную.

5. Виртуализация аппаратного обеспечения. В данном случае виртуализация заключается в разбиении компонент аппаратного обеспечения на сегменты, управляемые отдельно друг от друга. В некоторых случаях,

виртуализация операционных систем невозможна без виртуализации аппаратного обеспечения.

6. Виртуализация систем хранения. В свою очередь делится на два типа: виртуализацию блоков и виртуализацию файлов. Виртуализация файлов, как правило используется в системах хранения, при этом ведутся записи о том, какие файлы и каталоги находятся на определенных носителях. Виртуализация файлов отделяет статичный указатель нахождения виртуального файла (C:\, к примеру) от его физического местоположения. Т.е. при запросе пользователем файла C:\file.doc решение виртуализации файлов отправит запрос к месту реального размещения файла. Виртуализация блоков. Используется в сетях распределенного хранения данных. Сервера — хранилища данных используют RAID-технологию. iSCSI интерфейс также использует блочную виртуализацию, позволяя операционной системе распределить виртуальное блочное устройство. Более подробную информацию о виртуализации систем хранения см. в п.№4 списка источников для самостоятельного изучения.

7. Виртуализация сервисов. По своей сути, виртуализация сервисов является объединением всех вышеуказанных типов виртуализации. Решение виртуализации сервисов позволяет работать с приложением вне зависимости от физического расположения его частей, объединяя и управляя их взаимодействием.

Приведенная выше типология рассматривает виртуализацию, в зависимости от части ИТ — инфраструктуры, в которой она применяется. Подходы к созданию интерфейсов между виртуальными машинами и системами виртуализации ресурсов также можно разделить на следующие типы:

- Полная виртуализация — технология, которая обеспечивает полную симуляцию базового оборудования, гостевая операционная система остается в нетронутом виде.
- Аппаратная виртуализация — технология, позволяющая запускать на одном компьютере (хосте)

несколько экземпляров операционных систем (гостевых операционных систем). При этом гостевые ОС независимы друг от друга и от аппаратной платформы. Аппаратная виртуализация представляет собой набор инструкций, облегчающих выполнение операций на аппаратном уровне, которое до этого могли выполняться только программно, при этом затрачиваются дополнительные программные ресурсы.

- Паравиртуализация — техника виртуализации, при которой гостевые операционные системы подготавливаются для исполнения в виртуализированной среде, для этих целей в ядро ОС вносят незначительные изменения. Для взаимодействия с гостевой операционной системой используется API — интерфейс.

4. Различные виртуальные машины

Все отличия существующих виртуальных машин, по сути, сводятся лишь к перечню поддерживаемых ими **операционных систем**, а так же **стоимости**. Наиболее распространены сегодня системы VirtualBox, Windows Virtual PC и VMWare. Чем же они отличаются?

4.1. ORACLE VirtualBox — универсальная бесплатная виртуальная машина

VirtualBox — очень простой, мощный и бесплатный инструмент для виртуализации, развивающийся благодаря поддержжке знаменитой корпорации ORACLE. Он распространяется бесплатно, с открытым исходным кодом. VirtualBox

позволяет устанавливать в качестве «гостевой» практически любую современную операционную систему, будь то Windows, MacOS или любой из многочисленных представителей семейства Linux.

Преимуществом VirtualBox является простой и понятный пользовательский интерфейс. Хорошо сделан перевод на русский язык. Все основные функции вынесены в виде кнопок под меню. Создание виртуальных машин выполняется с помощью пошагового мастера.

VirtualBox поддерживает работу с сетями, поэтому ваша виртуальная ОС сможет легко выйти в Интернет. Очень полезной является функция «снимков» операционной системы. Виртуальная машина записывает на винчестер «точки восстановления», к которым вы в любой момент можете откатить гостевую систему в случае возникновения ошибок или сбоев.

4.2. Windows Visual PC — виртуальная машина от Microsoft

Windows Virtual PC — виртуальная машина для работы только и исключительно с Windows. Установить на Visual PC операционную систему Linux или MacOS просто невозможно.

Visual PC позволяет запускать несколько разных копий Windows на одном компьютере. Поддерживается работа с операционными системами Microsoft разных поколений, в том числе с 64-битными.

Плюсом Visual PC является возможность задать, какая из запущенных виртуальных машин будет более приоритетной по сравнению с другими. При этом «хостовый» компьютер сможет в автоматическом режиме выделять под ее нужды большее количество ресурсов за счёт других виртуальных систем, если «гостевой» системе это необходимо.

Моноплатформенность виртуальной машины Visual PC является её главным недостатком, впрочем, если требуется тестировать только разные версии Windows, это не актуально. Некоторым недостатком можно считать менее функциональный и менее удобный чем в VirtualBox интерфейс. В остальном Visual PC вполне надёжный инструмент, позволяющий тестировать операционные системы Microsoft.

4.3. VMware Workstation — для серьёзных задач

VMware Workstation – мощная, платная, максимально-надёжная программа для виртуализации, которая поддерживает работу с Windows и Linux. Для виртуализации MacOS, данная машина не предназначена.

Благодаря высокой надёжности и широчайшей функциональности VMware Workstation часто используется не просто для тестирования, а даже для постоянной работы виртуальных машин в качестве серверов даже для бизнес-приложений, будь то фаервол, отеляющий сеть организации от Интернет или даже сервер какой-либо базы данных.

VMware Workstation можно очень гибко настраивать, включая множество параметров сетевых подключений для работы с интернетом. Система имеет собственный виртуальный 3D-ускоритель, который позволяет получить высокое качество графики.

Интерфейс VMware Workstation достаточно грамотно организован, поэтому освоиться со всем её богатым функционалом довольно легко. В программе полностью поддерживается русский язык.

Необходимо отметить, что у VMware Workstation есть бесплатный «младший брат» — VMWare Player. В отличие от версии Workstation, плеер не умеет создавать виртуальные машины, но позволяет запускать ранее созданные. Эта программа будет полезна в случаях тестирования, когда, к примеру, разработчик какой-либо автоматизированной системы передат её на ознакомление именно в виде образа виртуальной машины. Эта практика получает всё большее распространение, поскольку избавляет пользователя от необходимости разворачивать незнакомую программу самостоятельно.

Задания

1. Установить **ORACLE VirtualBox.**
2. Запустить программу на исполнение.
3. Создать виртуальную машину для установки ОС Windiws XP.
4. Укажите объем оперативной памяти 343МБ.
5. Создайте новый виртуальный жесткий диск (тип VDI).
6. Укажите формат хранения «Фиксированный виртуальный жесткий диск».
7. Размер жесткого диска должен быть 11ГБ.
8. Покажите результат преподавателю.

Контрольные вопросы

1. Что называется виртуальной машиной?
2. Какие преимущества у виртуальной машины? Какие недостатки?
3. Чем отличается системная виртуальная машина от процессорной?
4. Что такое абстракция и виртуализация?
5. Перечислите основные типы виртуализаций.
6. Какие существуют подходы к созданию интерфейсов между виртуальными машинами и системами виртуализации ресурсов?
7. Какие существуют виртуальные машины? В чем их отличие друг от друга?

Лабораторная работа № 6
Установка и настройка операционной системы Linux

Цель работы: Приобрести опыт установки операционной системы Linux.

План проведения занятия

1. Закрепить знания о работе с программой VirtualBox.
2. Создать виртуальную машину исходя из предоставленной информации о минимальных аппаратных требованиях предлагаемой к установке и изучению операционной системы (ОС).
3. Установить ОС на виртуальный компьютер. Разобрать процесс установки ОС на этапы.
4. Познакомиться с основными группами программ входящих в состав ОС.

Краткие теоретические сведения

Linux (полное название GNU/Linux, произносится «гну слэш ») — общее название UNIX-подобных операционных систем на основе одноимённого ядра и собранных для него библиотек и системных программ, разработанных в рамках проекта GNU [2].

GNU/Linux работает на PC-совместимых системах семейства Intel x86, а также на IA-64, AMD64, PowerPC, ARM и многих других.

К операционной системе GNU/Linux также часто относят программы, дополняющие эту операционную систему, и прикладные программы, делающие её полноценной многофункциональной операционной средой.

В отличие от большинства других операционных систем, GNU/Linux не имеет единой «официальной» комплектации. Вместо этого GNU/Linux поставляется в

большом количестве так называемых дистрибутивов, в которых программы GNU соединяются с ядром Linux и другими программами.

Дистрибутив — это не просто набор программ, а ряд решений для разных задач пользователей, объединённых едиными системами установки, управления и обновления пакетов, настройки и поддержки [2].

Самые распространённые в мире дистрибутивы [2]:

- Ubuntu — быстро завоевавший популярность дистрибутив, ориентированный на лёгкость в освоении и использовании.
- Linux Mint (англ. mint - мята) — дистрибутив операционной системы Linux. Изначально Linux Mint основывался на Ubuntu, впоследствии число его вариаций стало расти и появилась ветка дистрибутивов на основе Debian с репозиториями ветки testing по умолчанию. В каждой из сборок используется одна из популярных графических сред — Mate, Cinnamon (является оболочкой для среды рабочего стола GNOME, являющаяся ответвлением от кодовой базы GNOME Shell), KDE, Xfce и другие. Так как сборки Linux Mint в большинстве своём отличаются от Ubuntu лишь составом включённых в дистрибутив пакетов, то совместимость с Ubuntu очевидна, что признают и сами разработчики. Цель дистрибутива заключается в предоставлении конечному пользователю максимально простой системы, которая будет готова к работе сразу после установки и не потребует загружать наиболее популярное программное обеспечение, в том числе и дополнительные кодеки для воспроизведения популярных мультимедийных форматов. В остальном Linux Mint практически идентичен Ubuntu, в которую привносит новый оригинальный интерфейс преимущественно в зелёных тонах и небольшое количество собственных приложений (mintInstall, mintUpdate, mintMenu и др.), призванных упростить работу тем, кто только знакомится с Linux. За это часто в шутку ему дают определение «Linux Mint — это Ubuntu, только лучше».

- openSUSE — бесплатно распространяемая версия дистрибутива SuSE, принадлежащая компании Novell. Отличается удобством в настройке и обслуживании благодаря использованию утилиты YaST.
- Fedora — поддерживается сообществом и корпорацией RedHat, предшествует выпускам коммерческой версии RHEL.
- Debian GNU/Linux — международный дистрибутив, разрабатываемый обширным сообществом разработчиков в некоммерческих целях. Послужил основой для создания множества других дистрибутивов. Отличается строгим подходом к включению несвободного ПО.
- Mandriva — французско-бразильский дистрибутив, объединение бывших Mandrake и Conectiva.
- Archlinux — ориентированный на применение самых последних версий программ и постоянно обновляемый, поддерживающий одинаково как бинарную, так иустановку из исходных кодов и построенный на философии простоты KISS, этот дистрибутив ориентирован на компетентных пользователей, которые хотят иметь всю силу и модифицируемость Linux, но не в жертву времени обслуживания.

Помимо перечисленных, существует множество других дистрибутивов, как базирующихся на перечисленных, так и созданных с нуля и зачастую предназначенных для выполнения ограниченного количества задач.

В отличие от Microsoft Windows (Windows NT), Mac OS (Mac OS X) и коммерческих UNIX-подобных систем, GNU/Linux не имеет географического центра разработки. Нет и организации, которая владела бы этой системой; нет даже единого координационного центра. Программы для Linux — результат работы тысяч проектов. Некоторые из этих проектов централизованы, некоторые сосредоточены в фирмах. Многие проекты объединяют хакеров со всего света, которые знакомы только по переписке. Создать свой проект или присоединиться к уже существующему может любой и, в случае успеха,

результаты работы станут, известны миллионам пользователей. Пользователи принимают участие в тестировании свободных программ, общаются с разработчиками напрямую, что позволяет быстро находить и исправлять ошибки и реализовывать новые возможности.

История развития UNIX-систем. GNU/Linux является UNIX-совместимой, однако основывается на собственном исходном коде [2]. Именно такая гибкая и динамичная система разработки, невозможная для проектов с закрытым кодом, определяет исключительную экономическую эффективность GNU/Linux. Низкая стоимость свободных разработок, отлаженные механизмы тестирования и распространения, привлечение людей из разных стран, обладающих разным видением проблем, защита кода лицензией GPL — всё это стало причиной успеха свободных программ.

Конечно, такая высокая эффективность разработки не могла не заинтересовать крупные фирмы, которые стали открывать свои проекты. Так появились Mozilla, OpenOffice.org, свободный клон Interbase (Borland).

Ход работы

Системные требования Runtu Light 12.04:

- Процессор: Pentium 3, 700MHz;
- Оперативная память: 256 Mb (32-bit)
- Свободное дисковое пространство: 3 Гбайт HDD + 256 Мбайт для swap.
- Видеоадаптер: 64 MB памяти;
- Устройство чтения DVD-дисков

Создадим виртуальную машину, учитываем тип операционной системы, а также минимальные системные требования.

Загружаем предлагаемый образ для установки Linux Ubuntu

18 июля 2012 года была представлена стабильная версия Runtu 12.04 с рабочим окружением XFCE, основанная на пакетной базе Ubuntu 12.04 LTS и полностью с ним совместимая.

Runtu — это Российская модификация дистрибутива Ubuntu Linux. Основной идеей Runtu является простота и лёгкость Ubuntu, соединённая с качественной русской локализацией и набором необходимого программного обеспечения, готового к использованию сразу после установки, а также в режиме LiveCD (система работающая с компакт диска). Главное отличие Runtu от оригинальной версии Ubuntu состоит в предустановленных приложениях (на установочном LiveCD содержатся приложения, за которые проголосовали пользователи на форуме разработчиков) и полной поддержки русского языка.

Runtu XFCE 12.04 это LiveCD, который может быть использован для работы в этом режиме, например для проверки поддержки/работоспособности имеющегося оборудования, для ремонтно-восстановительных работ (всё программное обеспечение для этого имеется), быстрой установки системы на жесткий диск и прочее...

В сборке вместо имеющейся в репозиториях Ubuntu 12.04 версии XFCE 4.8, установлена XFCE 4.10из PPA «Xubuntu Developers» team. Помимо основных компонентов рабочего окружения и утилит конфигурации, имеется базовый набор пользовательских приложений (офисный пакет, Интернет-браузер, клиент обмена мгновенными сообщениями, аудио/видео плеер и пр.). После полной загрузки системы, можно приступить к изучению и использованию Runtu XFCE.

Если планируется установка Runtu на жесткий диск, то можно освободить один из разделов под установку, используя LiveCD можно просто скопировать необходимые файлы (например перед переустановкой Windows) и многое другое... При необходимости в редактировании разделов можно воспользоваться GParted (графическое приложение для управления дисками, мощный редактор разделов).

Графический установщик Runtu XFCE 12.04 практически не отличается от оригинального установщика Ubuntu 12.04 (единственное его отличие – это отсутствие презентации, показываемой в процессе установки). Запускается «Установщик» двойным кликом по соответствующему значку на рабочем столе.

Если какой-то раздел жесткого диска был смонтирован (присоединён), например при копировании с диска на диск, то будет выдан запрос.

Далее предстоит выбрать «Тип установки». Выбор типа установки зависит от многих причин... Поэтому какой из них использовать зависит целиком от потребностей пользователя.

Первый вариант предполагает автоматическую установку Runtu рядом с имеющейся на компьютере системой. Достаточно только определить размер отводимого под Runtu места на жестком диске.

Второй тип установки, это замена (в полностью автоматическом режиме) установленной на компьютере системы на Runtu:

Пока не нажата кнопка «Установить сейчас» все действия обратимы, после нажатия внесённые изменения записываются на диск.

Третий тип, «Другой вариант» предоставляет возможность самостоятельно сделать разметку диска (в очень простом и интуитивно понятном менеджере разделов).

Обычно (как правило) на диске имеется два раздела... Так как имена разделов диска не отображаются (диск C или D), ориентироваться придётся по их размеру и/или заполнению. Если предполагается использовать на компьютере параллельно две системы то есть два возможных варианта продолжения установки. Первый вариант предусматривает выделение дискового пространства под Runtu на имеющемся разделе, второй вариант удаление диска (например диска D) и создание на нём разделов под Runtu.

Для начала нужно настроить раздел оставляемый без изменений, задать файловую систему (ntfs) и назначить точку монтирования для диска с Windows (диск C).

Размер диска нужно оставить прежним.

Второй раздел диска можно «Изменить» или «Удалить» и затем «Добавить», для создания нового раздела. Первым создаётся раздел основной раздел «/», вторым (обычно, но не как правило) раздел подкачки.

Swap / Подкачка — механизм/процесс работы с виртуальной памятью, при котором отдельные фрагменты памяти / страницы памяти (обычно не активные, долго не использовались, в которых нет необходимости) перемещаются из оперативной памяти на жёсткий диск (используется в качестве виртуальной памяти), освобождая ОЗУ для загрузки других фрагментов памяти (ядро разгружает память от наименее востребованных страниц).

Размер основного/системного раздела зависит в основном от потребностей пользователя в программном обеспечении и играх, но в любом случае 7-10Гб для этого раздела будет вполне достаточно. Для раздела подкачки (swap), при наличии более 2Гб оперативной памяти будет достаточно и 500Мб (при необходимости можно будет добавить программно), при 512Мбоперативной памяти раздел может быть 1Гб.

Оставшееся место отдаётся под раздел /home (там будут расположены пользовательские директории).

Данный вариант разметки диска полностью удовлетворяет потребностям большинства пользователей (более «хитрые способы» разметки удел специалистов и большинству пользователей не нужны).

Если в компьютере имеется два жестких диска (физических), то и для второго диска и разделов на нём нужно задать точки монтирования (ну и/или для диска D, для наличия доступа к нему). Для него можно задать произвольную точку монтирования (например /DickG), при этом он должен монтироваться в «/» или /media (например /media/DickG), а так

же задать файловую систему для него (как и с диском С, а при необходимости, поставив галочку, диск/раздел диска может быть форматирован).

Если диск не один, то есть возможность выбора диска для установки системного загрузчика (можно и даже нужно, оставить тот что определён автоматически).

По завершении настройки диска, ещё раз убедившись, что всё сделано правильно (так как хочется/нужно), все ранее сделанные изменения обратимы, нажатием на кнопку «Установить сейчас» запускается процесс установки системы.

Процесс установки запущен, далее необходимо задать дополнительные, пользовательские параметры.

Указать часовой пояс и используемую «по умолчанию» раскладку клавиатуры (при выборе русской раскладки, английская раскладка клавиатуры будет добавлена автоматически).

Завершающий шаг, создания пользователя (он же администратор системы), имени компьютера, пароля для пользователя, способа авторизации в системе и определение необходимости шифрования домашней папки пользователя.

О завершении установки известит диалог.

Если установка производилась с LiveCD, то можно продолжить знакомство с Runtu (нужно помнить, что все сделанные изменения после перезагрузки будут утеряны), а можно сразу перезагрузиться в установленную систему.

После авторизации и полной загрузки можно приступать к настройке, изучению и использованию Runtu.

В составе Runtu XFCE имеются все, графические утилиты для настройки рабочего окружения. Большинство из них доступно из общего интерфейса «Настройки» и из одноимённой строки системного меню.

В качестве файлового основного менеджера используется расширяемый плагинами, имеющий простой и интуитивный интерфейс файловый менеджер Thunar (являющийся частью окружения рабочего стола Xfce и по умолчанию не содержащий бесполезных или вносящих

путаницу элементов), так же имеется и MC / Midnight Commander (полнофункциональный и простой консольный файловый менеджер).

Легко настроить порядок загрузки систем (при включении компьютера выбирать одну из них для загрузки) поможет Grub Customizer.

В составе Runtu XFCE помимо стандартной имеется ещё несколько тем оформления (из состава XFCE). Если имеющиеся темы не удовлетворяют, то на сайте Xfce-Look.org можно всегда найти то что удовлетворит практически любого пользователя.

В Xfce используется оконный менеджер Xfwm, который включает в себя собственный композитный менеджер, с интересными эффектами окон, тенью, прозрачностью и прочим... Поддержка композитности может быть включена в дополнительных настройках «Диспетчера окон», в вкладке «Эффекты» (функция требует аппаратной поддержки графики). Для доступа к более тонким настройкам композитного менеджера (которые отсутствуют в официальном интерфейсе), можно воспользоваться Xfce4-Composite-Editor:

Для дальнейшего наращивания системы "под себя" и обновления установленного программного обеспечения можно использовать менеджер пакетов Synaptic.

Если есть желание/необходимость то в Synaptic можно установить «Центр приложений Ubuntu» (Ubuntu Software Center), для получения более визуально-понятного способа установки программного обеспечения и игр.

Можно использовать и менее требовательный к системным ресурсам «Центр приложений Lubuntu» (Lubuntu Software Center), так же доступный для установки в Synaptic.

Ubuntu Tweak поможет не только настроить систему и установить приложения (отсутствующие в основных источниках пакетов), но и очистить систему от ставших ненужными пакетов, конфигурационных файлов, старых версий ядер и прочего...

Так же можно использовать Ailurus (графическое приложение для тонкой настройки систем Ubuntu), несомненно очень полезный начинающим пользователям инструмент, предоставляющий лёгкий доступ к множествам настроек Runtu XFCE из одного места.

Во многом (функционально) приложение очень похоже на Ubuntu Tweak но отличается от него тем, что больше ориентирован на предоставление легкого доступа к «скрытым настройкам системы» (тогда как Ubuntu Tweak предназначен скорее для установки программного обеспечения из ppa-репозиториев) и в любом случае Ailurus будет прекрасным дополнением к возможностям Ubuntu Tweak (Ailurus хоть и давно не обновлялся, но на данный момент он полностью работоспособен).

В состав Runtu XFCE входит веб-браузер Mozilla Firefox, но на компьютерах с оперативной памятью 1Гб и менее лучше всё же использовать менее требовательный к системным ресурсам веб-браузер, например набор Интернет приложений, в том числе веб-браузер SeaMonkey (это не обязательно к исполнению, это моё личное мнение).

Установить SeaMonkey можно выполнив в консоли:

```
sudo add-apt-repository ppa:joe-nationnet/seamonkey-dev
sudo apt-get update
sudo apt-get install seamonkey
```

SeaMonkey — проект основан на добрых традициях Mozilla Suite, целью которого было совместить интегрированную структуру Mozilla Suite с новыми возможностями официальных программ Mozilla Firefox и Mozilla Thunderbird. SeaMonkey имеет более богатые настройки, чем Firefox и Thunderbird вместе взятые, а для удобства работы с ним требуется намного меньше расширений (во многом аналогичных расширениям Firefox). Это положительным образом сказывается на стабильности и потреблении системных ресурсов...

Минимальные системные требования Runtu XFCE не велики, для комфортной работы в системе достаточно иметь

3Гб свободного места на HDD (для установленной системы, а для работы в режиме LiveCD жесткий диск не требуется), процессор уровня P3 700Мгц и 256Мб оперативной памяти...

Но наиболее стабильно и комфортная работа с графическими приложениями достигается при использовании оперативной памяти 512Мб и более (при установке на ПК с ОЗУ менее 512Мб желательно устанавливать систему сразу, без загрузки LiveCD).

Контрольные вопросы

1. В чем состоит сущность системы Linux?
2. Что такое дистрибутив?
3. Перечислите основные дистрибутивы Linux. Объясните в чем их отличие.
4. Какую файловую систему использует для работы установленный Вами дистрибутив?
5. Перечислите основные этапы установки операционной системы.
6. Каким образом осуществляется установка пакетов приложений в системах Linux?
7. Что входит в состав установленного дистрибутива?

Лабораторная работа № 7
Терминал и командная оболочка операционной системы Linux

Цель работы: Приобрести опыт работы с командной строкой ОС Linux, изучить основные команды (рабочая станция, рабочий директорий, пользователи, дата, календарь, список процессов, завершение работы).

План проведения занятия

- Ознакомиться с краткими теоретическими сведениями.
- Приобрести навыки работы в терминале Linux. Научиться создавать новых пользователей при помощи терминала Linux, задавать несложные команды.

Краткие теоретические сведения

Стандартные команды в Linux отличаются от команд DOS и Windows — обычно они короче. При работе с командной строкой как обычно мигающий курсор обозначает позицию ввода текста, командная строка начинается с текущего пути и имени компьютера, за которым следует символ $, % или #. Последний означает, что команды будут выполняться от имени суперпользователя root. Символ ~ означает путь к текущей домашней директории пользователя.

Большинство команд в Linux, не требующих вывода информации пользователю, в случае успешного завершения вообще ничего не выводят на экран. Выводятся только ошибки и предупреждения в случае нарушения нормального выполнения команды. Т.е. в Linux действует общий принцип «молчит, значит работает».

В любом терминале Linux стрелками вверх/вниз на клавиатуре можно листать историю команд, которая сохраняется между сеансами работы и различается для разных

пользователей и хостов. Набранное частично команда, или имя файла, или каталога в текущей директории может быть автоматически дописано клавишей TAB. Если найдено более одного варианта и однозначно продолжить команду по TAB невозможно, то выводятся все подходящие варианты.

При работе в графической среде удобны эмуляторы терминала. Как правило они поддерживают закладки — несколько терминалов в одном окне, поддерживают цветовые схемы. Наиболее распространены эмуляторы терминала Gnome Terminal, Konsole, XFCE Terminal.

Терминал — эмулятор консоли. Именно в терминале мы будем работать с CLI (интерфейсом командной строки). Терминал часто также называют консолью или шеллом (от англ. shell — оболочка). В будущем для объяснения я буду использовать все три эти понятия, главное не забывайте, что они синонимы.

Многие пользователи и в особенности администраторы серверов под Linux в работе используют именно консоль, а не графическую оболочку, это связано с тем, что настройка и конфигурация Linux в основном заключается, в редактировании текстовых конфигурационных файлов. Даже если вы являетесь простым пользователем ОС Linux, большинство инструкций по настройке написаны с использованием консоли и знать основные команды жизненно необходимо.

Стоит обратить внимание на системные каталоги ОС в которых находятся файлы, необходимые для управления и сопровождения системы, а также стандартные программы. Их имена, расположение и содержание почти одинаковы почти во всех ОС Linux, поэтому эти каталоги называют также стандартными. Впрочем, на данный момент эпитет «стандартные» отражает скорее благие пожелания, чем действительность: иерархия каталогов одинакова только для дистрибутивов, связанных единством происхождения, а исторически сложившиеся различия создают опасность несовместимости разных дистрибутивов.

Краткое описание основных каталогов сведено в табл. 1.

Таблица 1

Основные каталоги [2]

Каталог	**Назначение**
/bin	Основные программы, необходимые для работы в системе: командные оболочки, файловые утилиты и т.п.
/sbin	Команды для системного администрирования, а также программы, выполняемые входе загрузки
/boot	Файлы, необходимые для загрузки системы (образ ядра)
/home	Домашние каталоги пользователей, кроме root
/dev	Файлы устройств
/etc	Файлы настроек: стартовые сценарии, конфигурационные файлы графической системы и различных приложений
/lib	Системные библиотеки, необходимые для основных программ, и модули ядра
/lost+found	Восстановленные после аварийного размонтирования части файловой системы
/media	Сюда обычно монтируются съемные носители: компакт-диски, flash-накопители
/mnt	Временные точки монтирования жестких дисков. Использовать этот каталог необязательно: подмонтировать файловую систему можно к любому другому каталогу
/opt	Дополнительные пакеты программ. Если программа, установленная сюда, больше не нужна, то достаточно удалить ее каталог без обычной процедуры деинсталляции
/proc	Виртуальная файловая система, дающая доступ к информации ядра. Другие файлы в этом каталоге в каждый момент времени содержат информацию о выполняющихся в этот момент программах

Продолжение табл. 1

Каталог	Назначение
/root	Домашний каталог суперпользователя. Домашние каталоги всех остальных могут находиться на отдельном разделе, но /root должен быть в корневой файловой системе, чтобы администратор всегда мог войти в систему для ремонтных работ
/tmp	Временные файлы
/var	Часто меняющиеся данные: системные журналы и протоколы приложений,замки, почтовые ящики, очереди печати и т.п.
/usr	Практически все остальное: программы, исходные коды, документация. Сюда по умолчанию устанавливаются новые программы

С точки зрения UNIX-подобных ОС, файл представляет собой поток или последовательность байтов [3]. Такой подход позволяет распространить понятие файла на множество ресурсов не только локального компьютера, но и удаленного, связанного с локальным сетью любого рода. Доступ к любому такому ресурсу осуществляется через универсальный интерфейс, благодаря чему запись данных в файл, отправка их на физическое устройство или обмен ими с другой работающей программой происходит аналогично. Это очень упрощает организацию данных и обмен ими.

В ОС Linux можно выделить следующие типы файлов:

- обычные файлы — последовательность байтов (текстовые документы, исполняемые программы, библиотеки и т.п.);
- каталоги — именованные наборы ссылок на другие файлы;
- файлы физических устройств, подразделяющихся на:

1. файлы блочных устройств, драйверы которых буферизуют ввод-вывод с помощью ядра и файлы байт-ориентированных, или символьных, устройств, позволяющих

связанным с ними драйверам выполнять буферизацию собственными средствами;

2. символические ссылки (symlink, symbolic link);
3. именованные каналы (named pipes);
4. гнезда (sockets).

Ход работы

Для выполнения данной работы будем использовать ранее установленный Linux Ubuntu. Запускаем Linux. После прохождения идентификации включаем терминал.

Для работы в терминале Ubuntu требуются права пользователя root, но, к сожалению, по умолчанию, он недоступен, поэтому для выполнения некоторых (не всех) команд надо писать sudo <команда>, и подтверждать свои права вводом пароля. И не пугайтесь того, что его не видно в терминале! Наберите точно по памяти, по окончании ввода нажмите Enter.

Для получении справки о дополнительных возможностях некоторых программ следует набрать <команда> —help

Потренируйтесь в выполнении команд:

- date
- oclock
- finger
- hwclock
- uname
- history
- clear
- ls

Создайте нового пользователя, при помощи терминала Ubuntu, и введите его в группу adm. Создайте пароль пользователю. Войдите под ним в систему. Процесс создания и ввода в группу внесите в отчет.

Разберите выполнение незадействованных команд табл. 2. Потренируйтесь в выполнении, определите их назначение и область применения.

Таблица 2

Команды

Команда	**Описание команды**
halt	стремительное и корректное выключение системы.
poweroff	корректное выключение системы.
reboot	корректное выключение с последующей загрузкой.
adduser	создание нового пользователя.
date	показывает нынешние дату и время, по системным часам ядра.
oclock	обычные часы
finger	отображение информации о пользователе
hostname	команда показывает личный номер этого узла сети
hwclock	интегрированные часы
uname	выводит информацию об используемой операционной системе
uptime	проявляет текущее время, длительность сеанса, число пользователей и загруженность процессора.
usermod	изменение параметров пользователя.
users	отражает короткий перечень пользователя работающих в системе в этот эпизод
whoami	демонстрирует нынешний личный номер пользователя, работающего в этом терминале.
write	посылает известные иному пользователя, окружающему в системе, маршрутом копирования строчек с терминала отправителя на терминал получателя.
history	демонстрирует пронумерованный перечень команд, которые Вы исполняли в данном и прошлом сеансе. Само собой разумеется, что если в перечне истории их очень немало, то увидите заключительные.
passwd	изменение пароля пользователя

Продолжение табл. 2

Команда	Описание команды
ps	выводит перечень всех работающих действий
times	проявляет абсолютное время исполнения действий для всей системы и этого пользователя
free	отражает информацию о своевременной памяти, подкачки, кэше, свободная память, общественная и т.п.
ls	указывает все файлы в текущем каталоге в алфавитном порядке. По всей вероятности аналогична dir
clear	чистит экран терминала (в случае если данное вполне вероятно)
ifconfig	отражает состояние текущей конфигурации сети или же настраивает сетевой интерфейс
less	отражает содержимое указанного файла на экране и дозволяет комфортно просмотреть
mkpasswd	создает качественный пароль, состоящий по умолчанию из 9 знаков и имеющий как минимум буквы в различном регистре и числа

Контрольные вопросы

1. Что такое терминал?
2. Перечислите основные команды терминала.
3. Чем отличаются командная строка Windows от терминала Linux? Чем отличаются команды?
4. Возможно ли полноценное управление Linux, используя только терминал? Ответ обоснуйте.
5. Перечислите основные системные каталоги.
6. Расскажите о типах файлов в ОС Linux.
7. Что представляет из себя файл в Linux и чем подобная схема отличается от Windows?

Лабораторная работа № 8
Изучение файловой системы ОС Linux и функций по обработке и управлению данными

Цели работы:

- Изучение команд, связанных с пользователями и группами;
- Изучение структуры файловой системы Linux, изучение общих команд создания, удаления, модификации файлов и каталогов, функций манипулирования данными, алиасами;
- Изучение иерархии процессов;
- Приобретение навыков по смене атрибутов объектов, смене прав доступа к объектам;
- Изучение организации безопасности системы, местонахождение файлов с паролями, просмотр системной информации и получение служебной информации.

План проведения занятия

1. Ознакомиться с краткими теоретическими сведениями. Приобрести навыки работы в терминале Linux.
2. Научиться создавать новые файлы и каталоги, разобрать назначение прав доступа к файлам и папкам.
3. Приготовиться к сдаче лабораторной работы.

Краткие теоретические сведения

Файловая структура системы Linux

В операционной системе Linux файлами считаются обычные файлы, каталоги, а также специальные файлы, соответствующие периферийным устройствам (каждое устройство представляется в виде файла). Доступ ко всем файлам однотипный, в том числе, и к файлам периферийных устройств. Такой подход обеспечивает независимость

программы пользователя от особенностей ввода/вывода на конкретное внешнее устройство.

Файловая структура Linux имеет иерархическую древовидную структуру. В корневом каталоге размещаются другие каталоги и файлы, включая 5 основных каталогов [2]:

bin — большинство выполняемых командных программ и *shell* — процедур;

tmp — временные файлы;

usr — каталоги пользователей (условное обозначение);

etc — преимущественно административные утилиты и файлы;

dev — специальные файлы, представляющие периферийные устройства; при добавлении периферийного устройства в каталог /dev должен быть добавлен соответствующий файл (черта / означает принадлежность корневому каталогу).

Текущий каталог — это каталог, в котором в данный момент находится пользователь. При наличии прав доступа, пользователь может перейти после входа в систему в другой каталог. Текущий каталог обозначается точкой (**.**); родительский каталог, которому принадлежит текущий, обозначается двумя точками (**..**).

Полное имя файла может включать имена каталогов, включая корневой, разделенных косой чертой, например: /home/student/file.txt. Первая косая черта обозначает корневой каталог, и поиск файла будет начинаться с него, а затем в каталоге home, затем в каталоге student.

Один файл можно сделать принадлежащим нескольким каталогам. Для этого используется команда **ln (link)**:

ln <имя файла 1> <имя файла 2>

Имя 1-го файла — это полное составное имя файла, с которым устанавливается связь; имя 2-го файла — это полное имя файла в новом каталоге, где будет использоваться эта связь. Новое имя может не отличаться от старого. Каждый файл может иметь несколько связей, т.е. он может

использоваться в разных каталогах под разными именами. Команда **ln** с аргументом -s создает символическую связь:

ln -s <имя файла 1> <имя файла 2>

Здесь имя 2-го файла является именем символической связи. Символическая связь является особым видом файла, в котором хранится имя файла, на который символическая связь ссылается. Linux работает с символической связью не так, как с обычным файлом — например, при выводе на экран содержимого символической связи появятся данные файла, на который эта символическая связь ссылается.

В Linux различаются 3 уровня доступа к файлам и каталогам [2]:

1) доступ владельца файла;
2) доступ группы пользователей, к которой принадлежит владелец файла;
3) остальные пользователи.

Для каждого уровня существуют свои байты атрибутов, значение которых расшифровывается следующим образом:

r – разрешение на чтение;

w – разрешение на запись;

x – разрешение на выполнение;

— – отсутствие разрешения.

Первый символ байта атрибутов определяет тип файла и может интерпретироваться со следующими значениями:

— – обычный файл;

d – каталог;

l – символическая связь;

в – блок-ориентированный специальный файл, который соответствует таким периферийным устройствам, как накопители на магнитных дисках;

с – байт-ориентированный специальный файл, который может соответствовать таким периферийным устройствам как принтер, терминал.

В домашнем каталоге пользователь имеет полный доступ к файлам (READ, WRITE, EXECUTE; r, w, x).

Атрибуты файла можно просмотреть командой **ls -l** и они представляются в следующем формате:

```
d          rwx          rwx          rwx
|           |            |            |
|           |            |            | Доступ для остальных
|           |            | Доступ к файлу для членов группы
|           | Доступ к файлу владельца
| Тип файла (директория)
```

Пример. Командой **ls -l** получим листинг содержимого текущей директории student:

```
— rwx — — 2 student 100  Mar  10 10:30 file_1
— rwx — r— 1 adm      200  May 20 11:15 file_2
— rwx — r— 1 student 100  May 20 12:50 file_3
```

После байтов атрибутов на экран выводится следующая информация о файле:

— число связей файла;

— имя владельца файла;

— размер файла в байтах;

— дата создания файла (или модификации);

— время;

— имя файла.

Атрибуты файла и доступ к нему, можно изменить командой:

chmod <коды защиты> <имя файла>

Коды защиты могут быть заданы в числовом или символьном виде. Для символьного кода используются:

знак плюс (+) — добавить права доступа;

знак минус (-) — отменить права доступа;

r,w,x — доступ на чтение, запись, выполнение;

u,g,o — владельца, группы, остальных.

Коды защиты в числовом виде могут быть заданы в восьмеричной форме. Для контроля установленного доступа к своему файлу после каждого изменения кода защиты нужно проверять свои действия с помощью команды **ls -l.**

Примеры

chmod g+rw,o+r file.1 — установка атрибутов чтения и записи для группы и чтения для всех остальных пользователей;

ls -l file.1 — чтение атрибутов файла;

chmod o-w file.1 — отмена атрибута записи у остальных пользователей;

>letter — создание файла letter. Символ > используется как для переадресации, так и для создания файла;

cat — вывод содержимого файла;

cat file.1 file.2 > file.12 — конкатенация файлов (объединение);

mv file.1 file.2 — переименование файла file.1 в file.2;

mv file.1 file.2 file.3 directory — перемещение файлов file.1, file.2, file.3 в указанную директорию;

rm file.1 file.2 file.3 — удаление файлов file.1, file.2, file.3;.

cp file.1 file.2 — копирование файла с переименованием;

mkdir namedir — создание каталога;

rm dir_1 dir_2 — удаление каталогов dir_1 dir_2;

ls [acdfgilqrstv CFR] namedir — вывод содержимого каталога; если в качестве namedir указано имя файла, то выдается вся информация об этом файле.

Значения аргументов:

— l –- список включает всю информацию о файлах;

— t – сортировка по времени модификации файлов;

— a – в список включаются все файлы, в том числе и те, которые начинаются с точки;

— s – размеры файлов указываются в блоках;

— d – вывести имя самого каталога, но не содержимое;

— r – сортировка строк вывода;

— i – указать идентификационный номер каждого файла;

— v – сортировка файлов по времени последнего доступа;

— q – непечатаемые символы заменить на знак ?;

— c – использовать время создания файла при сортировке;

— g – то же что -l, но с указанием имени группы пользователей;

— f – вывод содержимого всех указанных каталогов, отменяет флаги -l, -t, -s, -r и активизирует флаг -a;

— C – вывод элементов каталога в несколько столбцов;

— F – добавление к имени каталога символа / и символа * к имени файла, для которых разрешено выполнение;

— R – рекурсивный вывод содержимого подкаталогов заданного каталога.

cd <namedir> — переход в другой каталог. Если параметры не указаны, то происходит переход в домашний каталог пользователя.

pwd — вывод имени текущего каталога;

grep [-vcilns] [шаблон поиска] <имя файла> — поиск файлов с указанием или без указания контекста (шаблона поиска).

Значение ключей:

— v – выводятся строки, не содержащие шаблон поиска;

— c – выводится только число строк, содержащих или не содержащих шаблон;

— i – при поиске не различаются прописные и строчные буквы;

— l – выводятся только имена файлов, содержащие указанный шаблон;

— n – перенумеровать выводимые строки;

— s – формируется только код завершения.

Примеры

1. Напечатать имена всех файлов текущего каталога, содержащих последовательность «student» и имеющих расширение .txt:

grep -l student *.txt

2. Определить имя пользователя, входящего в ОС Linux с терминала tty23:

who | grep tty23

Порядок выполнения работы

1. Ознакомиться с файловой структурой ОС Linux. Изучить команды работы с файлами.
2. Используя команды ОС Linux, создать два текстовых файла.
3. Полученные файлы объединить в один файл и его содержимое просмотреть на экране.
4. Создать новую директорию и переместить в нее полученные файлы.
5. Вывести полную информацию обо всех файлах и проанализировать уровни доступа.
6. Добавить для всех трех файлов право выполнения членам группы и остальным пользователям.
7. Просмотреть атрибуты файлов.
8. Получить информацию об активных процессах и имена других пользователей.

Контрольные вопросы

1. Файловая структура Linux. Дайте характеристику.
2. Что считается файлами в ОС Linux?
3. Объясните назначение связей с файлами и способы их создания.
4. Что определяет атрибуты файлов и каким образом их можно просмотреть и изменить?
5. Какие методы создания и удаления файлов, каталогов Вы знаете?
6. В чем заключается поиск по шаблону?
7. Какой командой можно получить список работающих пользователей и сохранить его в файле?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Смит, Дж. Архитектура виртуальных машин [Текст] / Дж. Смит, Р. Наир. - Открытые системы. СУБД : журнал для профессионалов в области информационных технологий. - М. : Открытые системы. – 2005. – № 5-6

2. Официальная пользовательская документация Ubuntu Linux [Электронный ресурс] – Режим доступа: http://help.ubuntu.ru/wiki/linux

3. Бах, М. Дж. Архитектура операционной системы UNIX [Электронный ресурс] / М. Дж. Бах; пер. с англ. А.В. Крюков. – Режим доступа: http://www.lib.ru/BACH/

4. Гордеев, А. В. Операционные системы [Текст]: учебник для вузов / А. В. Гордеев. – 2-е изд. – СПб.: Питер, 2007.

5. Таненбаум Э. Современные операционные системы [Текст] / Э. Таненбаум. – 2-е изд. – СПб. : Питер, 2006.

6. Олифер, В. Сетевые операционные системы [Текст]: учебник для вузов / В. Олифер, Н. Олифер. – СПб.: Питер, 2005. – 544 с.

ОГЛАВЛЕНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам № 5-8 по дисциплинам
«Защита в операционных системах»,
«Безопасность операционных систем»
для студентов специальностей
090301 «Компьютерная безопасность»,
090303 «Информационная безопасность
автоматизированных систем»
очной формы обучения

Составители:
Савинков Андрей Юрьевич
Ленков Никита Александрович
Деревянко Владимир Николаевич

В авторской редакции

Подписано к изданию 27.04.2015.
Уч.-изд. л. 2,5.

ФГБОУ ВПО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14